UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/945,448	08/31/2001	Stephan Brunner	05306.P035	3554

7590 01/30/2006

Andre M. Gibbs
Blakely, Sokoloff, Taylor & Zafman LLP
Seventh Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025-1030

EXAMINER	
CHOW, CHIH CHING	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 01/30/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/945,448	BRUNNER ET AL.	
	Examiner	Art Unit	
	Chih-Ching Chow	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 October 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-78* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-78* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *04 January 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to amendment dated October 28, 2005.

2. Per Applicants' request, claims 1, 21, 41, and 59-78 have been amended. Claims 1-78 remain pending.

Response to Arguments

3. Applicant's arguments with respect to claims 1-78 have been considered but are moot in view of the new ground(s) of rejection necessitated by Applicant's amendments to the claims.

4. The examiner has reviewed the updated amendments, and noted that new matter has been introduced into the disclosure, therefore a new prior art has to be introduced. See 35 USC § 102 and 35 USC § 103 rejections (claims include the amendments) herein below:

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

6. Claims 1-14, 17, 18, 20-34, 37-38, 40-47, 49-51, 53-55, 57-72, 75-76, and 78 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,513,152 by Branson et al. (hereinafter "Branson").

CLAIM	Branson
1. A method for generating a configurator comprising: a. creating a customizable product class, the customizable product class including a set of one or more attributes to define the customizable product class.;	Branson teaches a method to generate a configurator, for item a. see Branson's column 2, lines 25-30, "According to the present invention, an object oriented framework mechanism for **customization of object oriented frameworks** provides an infrastructure that embodies the steps necessary to **customize a selected object oriented framework** (referred to herein as an

'input framework')"; and column 5, lines 56-60, "Our framework designer would next **design the classes and relationships** that make up the mechanisms shown on FIG. 1. A class is a definition of a set of like objects. As such, **a class can be thought of as an abstraction of the objects or as a definition of a type of object**." For item b, see Branson's column 12, lines 1-3, "A system that is modeled by an **object-oriented framework can be represented at a high level of abstraction by a diagram called a top-level class diagram**.(*customizable product class*)" And column 12, lines 6-10, "The boxes are arranged in a hierarchy such that boxes representing abstractions close to the physical: **components of the system** (*product components*) are at the lower levels of the diagram and boxes representing more abstract, functional **components** are closer to the top of the diagram." For item c, see Branson's FIG. 9 and column 16, lines 60 through column 17 line 6, "By providing framework mechanism 870 within computer system 800 to **customize input framework** 822, a uniform interface for all frameworks may be developed. Framework customization framework mechanism 870 may replace all of the disjoint proprietary applications that are currently used to **perform customization of different object oriented frameworks**. This allows a common **user interface for customizing** virtually any type of framework (*mapping a customizable UI to the customizable product class*). This **common user interface** greatly eases the burden of customizing object oriented frameworks. Thus, one of the primary benefits of the framework disclosed herein is the capability to perform framework **customization** using a simple, easy to use **user interface** defined by the framework customization framework". For item d, Branson's "customizable object oriented frameworks" are same as the "customizable products" in the current

b. adding a component product class to the customizable product class, the component product class is a subclass of the customizable product class; and

c. mapping a customizable UI to the customizable product class, the customizable UI to provide access structure to the configurator,

d. wherein the customizable product class is to represent a consumer product and the component product class is to represent one or

more components of the consumer product.	application; and Branson's "an object oriented framework class" is the representation of a customizable object oriented framework, which is the same as the customizable product class is to represent a consumer product; and the subclasses of a framework class is the same as the component product class.
2. The method of claim 1 wherein the component product class includes component product subclasses.	For the feature of claim 1 see claim 1 rejection. For the rest of claim 2 feature see claim 1, items a and b rejections.
3. The method of claim 1 wherein the component product class inherits the attributes of the customizable product class.	For the feature of claim 1 see claim 1 rejection. For the rest of claim 3 feature see Branson's column 8 lines 7-9, "Less general classes (e.g., the feeder class) are said to **inherit characteristics** from the more general class or classes" and column 12, lines 60 to column 13 line 7, "Another refinement of a simple association between two classes is a type referred to as an inheritance relationship. **Inheritance is a relationship among classes in which one class shares the structure and/or behavior associated with one or more other classes** (*inherit the attributes or cardinality attributes*). An inheritance association is also referred to as a "is a" relationship. Thus, given two classes A and B, the class A has an inheritance relationship with the class B if A is an example of a B; A is said to be a subclass of B and B is said to be a superclass or parent of A. That is, A "is a" B. An inheritance relationship is denoted with a connecting line that includes an arrowhead at one end to indicate a **subclass that derives its characteristics from a parent class** at the other end of the line."
4. The method of claim 1 further comprising: adding one or more component product classes to a port; and adding the port to the customizable product class, the port to allow the configurator to classify a group of component products.	For the feature of claim 1 see claim 1 rejection. For the rest of claim 4 feature see Branson's Fig.2A, component product classes can always be added to the product, each 'port' allows to classify a group of component products.

5. The method of claim 4 wherein the port includes a cardinality attribute, the cardinality attribute to constrain the number of component products to be added by the configurator.	For the feature of claim 4 see claim 4 rejection. For the rest of claim 5 feature see claim 3 rejection (attributes).
6. The method of claim 5 wherein the cardinality attribute includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.	For the feature of claim 5 see claim 5 rejection. For the rest of claim 6 feature see claim 3 rejection, the 'less general class' and 'more general class' part.
7. The method of claim 5 wherein the cardinality attribute includes a default cardinality, the default cardinality defines a quantity of the component product class added by the configurator.	For the feature of claim 5 see claim 5 rejection. For the rest of claim 7 feature see Branson's column 9, lines 24-30, "the framework designer has explicitly designed the get_temp_range( ) operation such that it is not a pure virtual operation definition. This means that get temp range( ) has been generically defined as a **default** operation (*default cardinality*). As such, it is considered a virtual operation. **Default** operations are used to provide generic function to subclasses (default operation for component product classes)."
8. The method of claim 1 wherein the mapping to include building the customizable UI from a set of themes, groups, and controls.	For the feature of claim 1 see claim 1 rejection. For the rest of claim 8 feature see claim 1 c rejection.
9. The method of claim 8 wherein the themes are tabs and wizards.	For the feature of claim 8 see claim 8 rejection. For the rest of claim 9 feature see Branson's FIG. 9 and column 17, lines 15-22, "The user selects from the extension points a subset of extension points to extend (step 930) (*tabs and wizards*). A subset in this context means any number of extension points from one up to and including the maximum number of extension points available in the input framework. Next, one of the extension points in the subset is selected (step 940). The user then interacts with the **extension point wizard** to complete

	the extension (step 945)."
10. The method of claim 8 wherein each theme in the set of themes, groups, and controls includes at least one of the set of background colors, fonts, and multi-linguals.	For the feature of claim 8 see claim 8 rejection. For the rest of claim 10 feature see Branson's column 23, lines 39-45, "The customizationController (*controls*) object then invokes its own displayFrameworkRepresentation( ) method (step 9), which presents a representation of theCustomizedFramework to theFrameworkCustomizer. In the case of a human user, the representation of theCustomizedFramework will be a **graphical representation** (*for a graphical representation, there must be background colors, fonts, and multi-linguals*)."
11. The method of claim 8 wherein the group includes one or more of the controls.	For the feature of claim 8 see claim 8 rejection. for the rest of claim 11 feature see claim 10 rejection.
12. The method of claim 8 wherein the control includes one or more of a drop down box, a radio button, and a list box.	For the feature of claim 8 see claim 8 rejection. For the rest of claim 12 feature see claim 1c rejection, the User Interfaces allows the user to customize his/her own control method.
13. The method of claim 1 wherein the customizable UI is used to generate a user interface for a component product class.	Same as claim 1 rejection.
14. The method of claim 1 wherein the customizable UI is a subclass of the customizable product.	See claim 1 rejection.
17. The method of claim 1 wherein the component product class, customizable class rules, and UI class are object oriented classes.	For the feature of claim 1 see claim 1 rejection.
18. The method of claim 1 wherein the customizable product has an object oriented structure.	For the feature of claim 1 see claim 1 rejection. Branson's disclosure has an Object Oriented structure.
20. The method of claim 1 wherein the configurator is stored in a data store.	For the feature of claim 1 see claim 1 rejection. For the rest of claim 20 feature see Branson's

	FIG. 8 (Mass storage) to store the configurator.
21. A machine-readable medium that provides instructions, which when executed by a set of one or more processors, cause the set of processors to perform operations for generating a configurator comprising: creating a customizable product class, the customizable product including a set of one or more attributes to define the customizable product; adding a component product class to the customizable product class, the component product class is a subclass of the customizable product; and mapping a customizable UI to the customizable product class, the customizable UI to provide access structure to the configurator, wherein the customizable product class is to represent a consumer product and the component product class is to represent one or more components of the consumer product.	Branson's disclosure also executed in a machine-readable medium (FIG. 8), see claim 1 rejection.
22. The machine-readable medium of claim 21 wherein the component product class includes component product subclasses.	For the feature of claim 21 see claim 21 rejection. For the rest of claim 22 feature see claim 2 rejection.
23. The machine-readable medium of claim 21 wherein the component product class inherits the attributes of the customizable product class.	For the feature of claim 21 see claim 21 rejection. For the rest of claim 23 feature see claim 3 rejection.
24. The machine-readable medium of claim 21 further comprising: adding one or more component product classes to a port; and adding the pod to the customizable product class, the port to allow the configurator to classify a group of component products,	For the feature of claim 21 see claim 21 rejection. For the rest of claim 24 feature see claim 4 rejection.
25. The machine-readable medium of claim 24 wherein the port includes a cardinality attribute, the cardinality attribute to constrain	For the feature of claim 24 see claim 24 rejection. For the rest of claim 25 feature see claim 5 rejection.

the number of component products to be added by the configurator.	
26. The machine-readable medium of claim 25 wherein the cardinality attribute includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.	For the feature of claim 25 see claim 25 rejection. For the rest of claim 26 feature see claim 6 rejection.
27. The machine-readable medium of claim 25 wherein the cardinality attribute includes a default cardinality, the default cardinality defines a quantity of the component product class added by the configurator.	For the feature of claim 25 see claim 25 rejection. For the rest of claim 27 feature see claim 7 rejection.
28. The machine-readable medium of claim 21 wherein the mapping to include building the customizable UI from a set of themes, groups, and controls.	For the feature of claim 21 see claim 21 rejection. For the rest of claim 28 feature see claim 8 rejection.
29. The machine-readable medium of claim 28 wherein the themes includes tabs and wizards.	For the feature of claim 28 see claim 28 rejection. For the rest of claim 29 feature see claim 9 rejection.
30. The machine-readable medium of claim 28 wherein the theme includes background color, fonts, and multi-lingual.	For the feature of claim 28 see claim 28 rejection. For the rest of claim 30 feature see claim 10 rejection.
31. The machine-readable medium of claim 28 wherein the group includes one or more of the controls.	For the feature of claim 28 see claim 28 rejection. For the rest of claim 31 feature see claim 11 rejection.
32. The machine-readable medium of claim 28 wherein the control includes one or more of a drop down box, a radio button, and a list box.	For the feature of claim 28 see claim 28 rejection. For the rest of claim 32 feature see claim 12 rejection.
33. The machine-readable medium of claim 21 wherein the customizable UI is used to generate a user interface for a component product class.	For the feature of claim 21 see claim 21 rejection. For the rest of claim 33 feature see claim 13 rejection.

34. The machine-readable medium of claim 21 wherein the customizable UI is a subclass of the customizable product.	For the feature of claim 21 see claim 21 rejection. For the rest of claim 34 feature see claim 14 rejection.
37. The machine-readable medium of claim 21 wherein the component product class, customizable class rules, and UI class are object oriented classes.	For the feature of claim 21 see claim 21 rejection. For the rest of claim 37 feature see claim 17 rejection.
38. The machine-readable medium of claim 21 wherein the customizable product has an object oriented structure.	For the feature of claim 21 see claim 21 rejection. For the rest of claim 38 feature see claim 18 rejection.
40. The machine-readable medium of claim 21 wherein the configurator is stored in a data store.	For the feature of claim 21 see claim 21 rejection. For the rest of claim 40 feature see claim 20 rejection.
41. An object oriented configurator comprising: a customizable product class; a component product, the component product is a subclass of the customizable product, the component product inherits a set of one or more attributes from the customizable product class; and a customizable UI, the customizable UI is mapped to the customizable product providing a view of the component product, wherein the customizable product class is to represent a consumer product and the component product class is to represent one or more components of the consumer product.	Branson's disclosure is also an objected oriented configurator, same as claim 1 rejection.
42. The object oriented configurator in claim 41 further comprising: a port, the pod comprising a set of one or more of the component products.	For the feature of claim 41 see claim 41 rejection. For the rest of claim 42 feature see claim 4 rejection.
43. The object oriented configurator in claim 42 wherein the port includes a cardinality, the cardinality to constrain the number of component products to add to the customizable	For the feature of claim 42 see claim 42 rejection. For the rest of claim 43 feature see claim 5 rejection.

product class.

44. The object oriented configurator in claim 43 wherein the cardinality includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.	For the feature of claim 43 see claim 43 rejection. For the rest of claim 44 feature see claim 6 rejection.
45. The object oriented configurator in claim 43 wherein the cardinality includes a default cardinality, the default cardinality defines a quantity of the component product class added by the configurator.	For the feature of claim 43 see claim 43 rejection. For the rest of claim 45 feature see claim 7 rejection.
46. The object oriented configurator in claim 41 the customizable class rule, and customizable UI are subclasses of the customizable product.	For the feature of claim 41 see claim 41 rejection. For the rest of claim 46 feature see claim 8 rejection.
47. The object oriented configurator in claim 41 wherein the component product includes a static attribute, the static attribute is not inherited from a parent class.	For the feature of claim 41 see claim 41 rejection. For the rest of claim 47 feature see claim 16 rejection.
49. The object oriented configurator in claim 41 further comprising a second customizable product.	For the feature of claim 41 see claim 41 rejection. Branson's disclosure allows to have multiple customizable product (the Zoo Administration is only an example).
50. The object oriented configurator in claim 49 wherein the component product includes one or more of a second customizable product.	See claim 49 rejection.
51. The object oriented configurator in claim 41 wherein the component product includes an expression to restrict the component product from becoming a subclass of the customizable product class.	For the feature of claim 41 see claim 41 rejection. Branson's disclosure also allows independent object class, which does not become a subclass of the customizable product class, see claim 1 rejection.
53. The object oriented configurator in claim 41 wherein the customizable UI includes a	For the feature of claim 41 see claim 41 rejection. For the rest of claim 53 feature see

theme, group, and control.	claim 8 rejection.
54. The object oriented configurator in claim 41 wherein the theme includes a tab, wizard, font, and color.	For the feature of claim 41 see claim 41 rejection. For the rest of claim 54 feature see claim 9 rejection.
55. The object oriented configurator in claim 41 wherein the control includes one or more of a drop down box, a radio button, and a list box.	For the feature of claim 41 see claim 41 rejection. For the rest of claim 55 feature see claim 12 rejection.
57. The object oriented configurator in claim 41 wherein each component product class has an unique identifier, the unique identifier is used to locate an associative component product.	For the feature of claim 41 see claim 41 rejection. For the rest of claim 57 see Branson's column 12, lines 32-35, "Each class is **identified by a name that is unique to the associated class** category and also indicates the relationship of each class to one of the mechanisms illustrated in FIG. 1."
58. The object oriented configurator in claim 41 further comprising link items.	For the feature of claim 41 see claim 41 rejection. For the rest of claim 58 see Branson's column 14, lines 1-3, "Objects and their interrelationships are represented in object diagrams that comprise object icons having links that indicate synchronization between objects".
59. An apparatus composed of logic blocks to customize a product comprising: a first logic block to create a customizable product class, the customizable product class including a set of one or more attributes to define the customizable product class. a second logic block to add a component product class to the customizable product class, the component product class is a subclass of the customizable product class; and a third logic block to map a customizable UI to the customizable product class, the customizable UI to provide access structure to the configurator, wherein the customizable oroduct class is to represent a consumer product and the component product class is to represent one or	Branson's disclosure also contains logic blocks, see Figs. 1, 4, and 5. See claim 1 rejection.

more components of the consumer product.

60. The apparatus of claim ~~1~~59 wherein the component product class includes component product subclasses.

For the feature of claim 59 see claim 59 rejection. For the rest of claim 60 feature see claim 2 rejection.

61 . The apparatus of claim ~~1~~59 wherein the component product class inherits the attributes of the customizable product class.

For the feature of claim 59 see claim 59 rejection. For the rest of claim 61 feature see claim 3 rejection.

62. The apparatus of claim ~~1~~ 59 further comprising:
a fourth logic block to add one or more component product classes to a port; and
a fifth logic block to add the port to the customizable product class, the port to allow the configurator to classify a group of component products.

For the feature of claim 59 see claim 59 rejection. For the rest of claim 62 feature see claim 4 rejection, Dardinski's disclosure allows more than one component product classes, and multiple logic blocks to add to the customizable product class.

63. The apparatus of claim 4 62 wherein the pod includes a cardinality attribute, the cardinality attribute to constrain the number of component products to be added by the configurator.

For the feature of claim 62 see claim 62 rejection. For the rest of claim 63 feature see claim 5 rejection.

64. The apparatus of claim ~~5~~ 63 wherein the cardinality attribute includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.

For the feature of claim 63 see claim 63 rejection. For the rest of claim 64 feature see claim 6 rejection.

65. The apparatus of claim ~~5~~63 wherein the cardinality attribute includes a default cardinality, the default cardinality defines a quantity of the component product class added by the configurator.

For the feature of claim 63 see claim 63 rejection. For the rest of claim 65 feature see claim 7 rejection.

66. The apparatus of claim ~~1~~59 wherein the third logic block to map to include building the customizable UI from a set of themes, groups, and controls.

For the feature of claim 59 see claim 59 rejection. For the rest of claim 66 feature see claim 8 rejection.

67. The apparatus of claim ~~8~~66 wherein the themes are tabs and wizards.	For the feature of claim 66 see claim 6 rejection. For the rest of claim 67 feature see claim 9 rejection.
68. The apparatus of claim ~~8~~66 wherein each theme in the set of themes, groups, and controls includes at Ieast one of the set of background colors, fonts, and multi-linguals.	For the feature of claim 66 see claim 66 rejection. For the rest of claim 68 feature see claim 10 rejection.
69. The apparatus of claim ~~8~~66 wherein the group includes one or more of the controls.	For the feature of claim 66 see claim 66 rejection. For the rest of claim 69 feature see claim 11 rejection.
70. The apparatus of claim ~~8~~66 wherein the control includes one or more of a drop down box, a radio button, and a list box.	For the feature of claim 66 see claim 66 rejection. For the rest of claim 70 feature see claim 12 rejection.
71. The apparatus of claim ~~1~~59 wherein the customizable UI is used to generate a user interface for a component product class.	For the feature of claim 59 see claim 59 rejection. For the rest of claim 71 feature see claim 13 rejection.
72. The apparatus of claim ~~1~~59 wherein the customizable UI is a subclass of the customizable product.	For the feature of claim 59 see claim 59 rejection. For the rest of claim 72 feature see claim 14 rejection.
75. The apparatus of claim ~~1~~59 wherein the component product class, customizable class rules, and UI class are object oriented classes.	For the feature of claim 59 see claim 59 rejection. For the rest of claim 75 feature see claim 17 rejection.
76. The apparatus of claim ~~1~~59 wherein the customizable product has an object oriented structure.	For the feature of claim 59 see claim 59 rejection. For the rest of claim 76 feature see claim 18 rejection.
78. The apparatus of claim ~~1~~59 wherein the configurator is stored in a data store.	For the feature of claim 59 see claim 59 rejection. For the rest of claim 78 feature see claim 20 rejection.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the

subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 15, 16, 35, 36, 56, 73 and 74 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,513,152 by Branson et al. (hereinafter "Branson"), in view of well-known technology (hereinafter "Well-Known").

CLAIM	Branson / Well-Known
15. The method of claim 1 wherein the customizable UI is used to generate a configurator user interface with HTML, Applets, and Activex programming languages.	For the feature of claim 1 see claim 1 rejection. HTML, Applets, and Activex are all well-known programming languages for implementing user interfaces.
16. The method of claim 1 wherein the component product class includes a static attribute, the static attribute is not associated with a parent class.	For the feature of claim 1 see claim 1 rejection. When an attribute is not defined under a customizable product (parent) class level, it's not going to inherit any attribute from its parent class. – a well-known Object Oriented programming language concept.
35. The machine-readable medium of claim 21 wherein the customizable UI is used to generate a configurator user interface with HTML, Applets, and ActiveX programming languages.	For the feature of claim 21 see claim 21 rejection. For the rest of claim 35 feature see claim 15 rejection.
36. The machine-readable medium of claim 21 wherein the component product class includes a static attribute, the static attribute is not associated with a parent class.	For the feature of claim 21 see claim 21 rejection. For the rest of claim 36 feature see claim 16 rejection.
56. The object oriented configurator in claim 41 wherein the customizable UI map comprises HTML, JAVA applets, and ActiveX components.	For the feature of claim 41 see claim 41 rejection. For the rest of claim 56 feature see claim 15 rejection.
73. The apparatus of claim ~~1~~59 wherein the customizable UI is used to generate a configurator user interface with HTML, Applets, and ActiveX programming languages.	For the feature of claim 59 see claim 59 rejection. For the rest of claim 73 feature see claim 15 rejection.
74. The apparatus of claim ~~1~~59 wherein the	For the feature of claim 59 see claim 59

component product class includes a static attribute, the static attribute is not associated with a parent class.	rejection. For the rest of claim 74 feature see claim 16 rejection.

11. Claims 19, 39, 48, 52, and 77 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,513,152 by Branson et al. (hereinafter "Branson"), in view of US Patent No. 6,754,885 by Darkinski et al.,(hereinafter "Dardinski").

CLAIM	**Branson / Dardinski**
19. The method of claim 1 wherein the customizable product includes versioning.	For the feature of claim 1 see claim 1 rejection. Branson teaches all the aspects of claim 19 but he does not mention 'versioning' specifically', however, Dardinski teaches 'versioning' in an analogous prior art. See Dardinski's Fig. 45, which depicts **version control** *(version control is used of versioning)* basic concepts in a system according to the invention". It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Branson's disclosure of the customization object oriented design by versioning the design taught by Dardinski for the purpose of providing the ability for the system to record changes made to the control database (Dardinski column 51, lines 60-61).
39. The machine-readable medium of claim 21 wherein the customizable product includes versioning.	For the feature of claim 21 see claim 21 rejection. For the rest of claim 39 feature see claim 19 rejection.
48. The object oriented configurator in claim 41 wherein the attribute is of type string, number, date, and Boolean.	For the feature of claim 41 see claim 41 rejection. For the rest of claim 48 feature see Dardinski's column 13, lines 8-9, "Data Specifies the data type of the Parameter. Integer, float, boolean, and Type string are examples of a data type".
52. The object oriented configurator in claim 41 further comprising: a script, the script to communicate with another application.	Branson teaches all aspects of claim 41, but he does not mention 'script' specifically, however Dardinski teaches 'script' in an analogous prior art. See Dardinski's column 39, lines 44-49, "this type of automation is typically referred to

	as '**Scripting**'. By exposing parameterized objects through automation and defining event interfaces, a **scripting engine** (such as VBScript) can be hooked into to run **event-based scripts**. This is a powerful tool for easily building and maintaining IDA functionality, as well as giving users an extremely rich and flexible way to customize and extend IDA" and lines 58-60, "An event is handled using a **script** (VBScript) that is persisted in a parameterized object and **passed with the object itself to the automation manager** *(using script to communicate with another application)*". It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Branson's disclosure of the customization object oriented design by versioning the design taught by Dardinski for the purpose of passing information with the object itself to the object manager (Dardinski column 39, lines 58-59).
77. The apparatus of claim ~~1~~59 wherein the customizable product includes versioning.	For the feature of claim 59 see claim 59 rejection. For the rest of claim 77 feature see claim 19 rejection.

Conclusion

The following summarizes the status of the claims:

35 USC § 102 rejection: Claims 1-14, 17-18, 20-34, 37-38, 40-47, 49-51, 53-55, 57-72, 75-76, and 78

35 USC § 103 rejection: Claims 15, 16, 19, 35, 36, 39, 48, 52, 56, 73, 74, and 77

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chih-Ching Chow whose telephone number is 571-272-3693. The examiner can normally be reached on 7:30am - 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any inquiry of a general nature of relating to the status of this application should be directed to the **TC2100 Group receptionist: 571-272-2100**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chih-Ching Chow

Examiner

Art Unit 2192

January 10, 2005

CC

TUAN DAM
SUPERVISORY PATENT EXAMINER